

UNIVERSIDAD DE CÓRDOBA

Reproductor de música web Jukloud

Prácticas de Redes

Manuel Pedrero Luque

30/05/2011

Contenido

Introducción	2
Justificación del trabajo.....	3
Requisitos	3
Diseño.....	4
Servidor	4
Desarrollo de escáner de archivos mp3 y base de datos	5
Desarrollo del servidor	5
Librerías adicionales	6
Fichero jkloud.py	6
Fichero mp3functions.py.....	6
Fichero webfunctions.py	7
Cliente	7
Fichero jukloud.html	7
Camino de datos.....	7
Pruebas.....	8
Futuras mejoras.....	8
Bibliografía	9
Apéndice A: Manual de Usuario.....	10
Bienvenido a Jukloud	10
Primeros pasos	10
El servidor de Jukloud	10
Escaneando la biblioteca de música con Jukloud	11
Utilizando la interfaz web de Jukloud	12
Accediendo a la interfaz desde el navegador	12
Compatibilidad de Jukloud	14
Jukloud en Firefox 4	15
Jukloud en Chrome 11.....	15
Jukloud en Safari versión iOS	16
Jukloud en Internet Explorer 9	16
Preguntas frecuentes	16

Introducción

En los últimos años la popularidad de escuchar nuestra música en la nube ha experimentado un crecimiento sin igual. Desde que servicios como LastFM o PandoraRadio abriesen el camino, esta tendencia no ha hecho más que aumentar. Surgieron servicios como Goeat o GrooveShark que iniciaron la filosofía de que fuese el propio usuario el que subiese su música a la nube para poderla escuchar después desde cualquier lugar. Mientras que el primero se asemeja a un Youtube orientado a música en lugar de a vídeo, el segundo intenta ofrecernos un reproductor web en el que todo el mundo puede colaborar y compartir su música.

Sin duda, la gran revolución llegó con Spotify, servicio que ofrecía un programa cliente a través del cual se tiene acceso a la que quizás sea la mayor biblioteca musical del mundo.

Sin embargo, no todo son ventajas con estos servicios: LastFM y Spotify están actualmente muy limitados a usuarios que no paguen una mensualidad para conseguir una cuenta Premium. Por otro lado GrooveShark, si bien parte de una gran idea, al final puede resultar incómodo de utilizar ya que precisamente esa libertad para compartir hace que el contenido existente no sea siempre de una calidad aceptable, y favorece el desorden.

Goeat por su parte es muy bueno para encontrar alguna canción puntual, pero no está pensado para escuchar discos enteros, y PandoraRadio lleva unos años vetada desde fuera de Estados Unidos por temas de licencias.

Una posible solución para esto puede ser usar un pequeño servidor de música online, que pueda transmitir en streaming las canciones a cualquier lugar, y un cliente lo más compatible posible con todas las plataformas, para no tener problema en acceder a nuestra música desde cualquier lugar. Por otro lado, es importante que sea cómodo y sencillo de manejar, ya que es en ese tema donde han fallado varios servicios en el pasado.

Con esta filosofía, y con la intención de aprender y aplicar los conocimientos adquiridos este año en la asignatura de Redes, surge la idea de desarrollar Jukloud (**Jukebox + Cloud**). Jukloud es un reproductor de música con una interfaz completamente web utilizando los últimos avances disponibles (HTML5, CSS3 y AJAX) con el objetivo de crear una interfaz sencilla, usable y cómoda, garantizando además la máxima compatibilidad posible con los dispositivos multimedia más utilizados hoy en día. De hecho, y aunque se prioriza el uso de HTML5, se ha incluido un pequeño reproductor Flash para mantener la compatibilidad con navegadores que no implementen las etiquetas de HTML5 utilizadas.

El servidor por su parte se ha desarrollado utilizando Python 2.7 y web.py, un framework ligero pero potente que en conjunción con la biblioteca JavaScript jQuery ha facilitado en gran medida toda la comunicación AJAX entre cliente y servidor, consiguiendo de este modo una interfaz más limpia y más cómoda.

Jukloud ha sido probado con éxito en Firefox 3.X y 4.X, Chrome/Chromium y Safari. Gracias al uso de HTML5 funciona perfectamente en iPhone y debería funcionar igualmente bien en Android. El único navegador que no soporta actualmente Jukloud es Internet Explorer, debido a su pobre soporte del estándar HTML5 y su baja compatibilidad con CSS y JavaScript.

Justificación del trabajo

Aunque el trabajo no se desarrolló utilizando Java (motivo por el cual se consultó antes al profesor), la realización del mismo ha sido muy provechosa en términos de aprendizaje y de aprovechamiento de la asignatura en opinión del alumno. Se han tenido que ver otras metodologías y modos de desarrollo distintos a los que se han estudiado tradicionalmente en otras asignaturas. Además, se han aprendido y utilizado varias tecnologías punteras tratadas en la asignatura, como el uso de AJAX para la comunicación cliente servidor de forma asíncrona y sin necesidad de recargar la página, el uso de las nuevas etiquetas de CSS3 y HTML5 para desarrollar tanto el diseño de la interfaz web como de la funcionalidad de la aplicación.

Por último, se han utilizado Frameworks (en este caso Web.py y jQuery) que proporcionan una capa mayor de abstracción que un desarrollo más tradicional como se ha hecho en prácticas de otras asignaturas.

Decir también que la elección de Python como lenguaje de desarrollo ha permitido al alumno conocer nuevos aspectos del mismo así como ver la potencia de Frameworks y librerías que pese a su etiqueta de ligeras han demostrado sus cualidades, siendo una de ellas la total compatibilidad con diferentes sistemas operativos, hasta el punto de que el trabajo, desarrollado bajo Debian, funciona perfectamente bajo Windows sin necesidad de tocar una sola línea de código, siendo en ese aspecto similar a Java.

Requisitos

Se detallan a continuación los requisitos propuestos por el alumno para la realización de la aplicación.

- | | |
|--------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| RF-1: | Debido a la gran popularidad del formato mp3, la aplicación deberá soportar dicho formato para evitar que el usuario final se vea obligado a convertir sus archivos de audio para utilizarla. |
| RF-2: | Con el objetivo de poner en práctica lo estudiado en la asignatura, la aplicación cliente se desarrollará utilizando HTML (4 y 5), CSS (2 y 3), JavaScript y el protocolo AJAX para las comunicaciones entre cliente y servidor. |
| RF-3: | El cliente que maneje el usuario deberá asemejarse lo máximo posible a un reproductor web de escritorio, permitiendo por ejemplo seleccionar listas de reproducción filtrando por artistas, álbumes o géneros, pausar y reanudar la reproducción, cambiar de pista, etc. Todas estas acciones se deben llevar a cabo de forma asíncrona, de manera que no sea necesario recargar la página web en ningún momento, dando así un aspecto más parecido a una aplicación de escritorio. |
| RF-4: | Se debe intentar maximizar la compatibilidad de la aplicación tanto en su parte web como en su parte servidora. Dado que HTML5 implementa métodos de fallback, deberán utilizarse para intentar proporcionar alternativas en caso de detectarse incompatibilidades con navegadores, especialmente a la hora de reproducir el audio. |
| RF-5: | El sistema debe tener una respuesta rápida aunque maneje muchas canciones. Por ello se diseñará una base de datos con la información de los ficheros de audio disponibles de manera que el tiempo de acceso a dicha información se minimice. |
| RF-6: | El sistema no debe estar limitado a un conjunto de ficheros de audio de prueba. Será necesario implementar rutinas de escaneo de ficheros para actualizar la información |

	de la base de datos ante nuevos ficheros de audio.
RF-7:	El sistema debe permitir el acceso simultáneo de varios usuarios a la aplicación web de manera independiente.
Rf-8:	Con el objetivo de maximizar la usabilidad de la aplicación, se debe trabajar con los metadatos ID3 de los ficheros mp3 y dar la posibilidad de realizar acciones comunes en los reproductores de escritorio como escuchar todas las pistas de un determinado artista o un género específico.
RF-9:	El sistema debe ser capaz de actualizar la base de datos de canciones y enviar las nuevas canciones a los usuarios sin que estos necesiten recargar la página, ni el administrador tenga que reiniciar el servicio.
RF-10:	Para asemejarse lo más posible a un reproductor de escritorio, dada una lista de reproducción, el reproductor debe cargar la siguiente pista al terminar la actual, y volver a la primera una vez terminada la lista de reproducción.

Diseño

Servidor

Para la creación del servidor, se barajaron diversas alternativas. La primera decisión importante fue la de utilizar un framework que facilitase algunas de las tareas que se tendrían que llevar a cabo, especialmente en cuanto a diseño web y comunicación cliente/servidor.

Se estudiaron tres Frameworks diferentes:

- **Django:** Es el framework más conocido para desarrollo web con Python. Es muy potente y completo, y con mucha documentación disponible y actualizaciones constantes por parte de los desarrolladores. Sin embargo, se descartó por ser demasiado *complejo* para la aplicación que se pretendía desarrollar. Su potencia era a la vez su mayor virtud pero también un defecto debido a la configuración previa y a su curva de aprendizaje.
- **CherryPy:** Es un framework reciente y muy activo, más ligero que Django pero con características bastante potentes, incluyendo un servidor web especialmente optimizado, soporte para las tecnologías y servidores más populares, sistemas de configuraciones y plugins, ejemplos, muy personalizable y con buen soporte.
- **Web.py:** Es un framework diseñado para aplicaciones web bastante simple y que se jacta de permitir usar el propio Python para casi todos los aspectos del desarrollo de la aplicación, incluyendo por ejemplo la posibilidad de incrustar código Python en las plantillas HTML que luego generarán las páginas. Una de sus características más sobresalientes es (según sus creadores) que no hace falta adaptarse a ningún modelo de desarrollo concreto, sino que deja total libertad para desarrollar siguiendo casi cualquier estilo. El propio Guido van Rossum (creador de Python) ha recomendado este framework en varias ocasiones.

Finalmente se optó por Web.py por su aparente sencillez, que puede verse en el apartado de ejemplos y *cookbook* de su web oficial y por la versatilidad que parecía ofrecer.

Desarrollo de escáner de archivos mp3 y base de datos

Se ha desarrollado un pequeño programa que recorre el sistema de archivos recursivamente desde la carpeta static/songs de la aplicación e introduce los datos de los archivos mp3 que encuentra en una pequeña base de datos en SQLite3. Esta base de datos es sencillamente una tabla que almacena los siguientes campos:

Campo	Descripción
Tracklist	Número de pista de la canción dentro de su disco.
Artist	Artista de la canción.
Álbum	Disco al que pertenece la canción.
Genre	Género al que pertenece la canción.
Title	Título de la canción.
Hash	Hash MD5 de cada canción, utilizado para identificar duplicados y no añadirlos de nuevo a la base de datos.
Path	Ruta relativa a cada canción. Utilizada para poderla establecer como src en la etiqueta de audio del reproductor.

A la hora de leer las etiquetas ID3 de los archivos mp3 se optó por la librería eyeD3, que además de ser muy sencilla de utilizar maneja los posibles fallos de lectura con excepciones, por lo que un fallo no se traduce en finalizar la aplicación, sino que se puede optar por saltarse la pista errónea y seguir escaneando.

Para tratar con la base de datos se utilizó la librería por defecto de Python (sqlite3) que permite crear cursores para realizar las consultas y las inserciones/borrados de forma muy cómoda.

Para obtener el hash MD5 se ha utilizado la librería hashlib, que proporciona métodos para obtener diferentes hashes de los archivos.

Por último, para recorrer los archivos y carpetas y obtener el path de los mismos se han utilizado las librerías os que proporciona métodos de comunicación independientes del sistema operativo, y urllib que permite obtener las URL de cada archivo y por tanto almacenarlos de una manera uniforme e independiente del sistema operativo de ejecución manteniendo la compatibilidad con la base de datos.

El programa de escaneo se encuentra en el fichero songlib.py

Desarrollo del servidor

El fichero principal del servidor es jukloud.py, mientras que los ficheros mp3functions.py y webfunctions.py implementan funciones auxiliares para recoger datos de la base de datos y generar los XML de respuesta necesarios para el uso de AJAX en las comunicaciones cliente-servidor.

Además son necesarias dos carpetas adicionales: la carpeta static contiene los elementos estáticos de la web, como los ficheros CSS, JavaScript, imágenes y las canciones de nuestra biblioteca. La carpeta templates contiene las plantillas de las páginas web dinámicas que generará el servidor.

Librerías adicionales

Se han utilizado dos librerías adicionales en JavaScript para conseguir un mayor grado de abstracción en el desarrollo de la aplicación:

- **jQuery.js:** para algunos efectos y transiciones y para facilitar las peticiones AJAX. Gracias al framework utilizado (web.py) es posible incluir directamente código Python en este fichero HTML. Dicho código es necesario escapararlo con \$. Es por ello que para las llamadas de jQuery se utiliza jQuery() en lugar de la función \$()
- **audio.js:** que proporciona un reproductor muy básico de audio utilizando las nuevas etiquetas de HTML5. Además, si detecta que el navegador no es compatible con ellas incluye una versión fallback en flash.

Fichero jkloud.py

En este fichero se desarrolla toda la parte del servidor. Para empezar, Web.py permite asignar URLs a clases, de modo que al cargar una URL determinada, se llamará al método GET de la clase que tenga asignada. El return de dicho método será el que determine la respuesta del servidor.

Se ha optado por dejar la URL vacía para el método main_player, que se encargará de construir la interfaz del reproductor, y luego establecer varias URLs del tipo /Ajax/* en la que a cada una se le corresponde la respuesta de una petición AJAX que haga el cliente a través del reproductor web.

Mediante la línea:

```
app = web.application(urls, globals(), True)
```

Se crea una instancia que sirve de mediadora entre la aplicación servidor y la web que entre otras cosas referencia las URLs definidas anteriormente y posibilita que los cambios que se hagan a la aplicación web se reflejen sin necesidad de reiniciar el servidor.

Mediante la línea:

```
render = web.template.render('templates/', base='base')
```

Se llama a un renderizador que se encarga de construir la web a partir de las plantillas del directorio templates.

El resto del fichero son las implementaciones de cada una de las URLs que han considerado en la aplicación. La única URL que verá el cliente será la raíz, siendo el resto respuestas de peticiones AJAX.

Fichero mp3functions.py

Este fichero contiene todos los métodos necesarios para realizar las peticiones de consulta a la base de datos y devolver los mismos en un formato adecuado para ser tratado internamente por Python. Es el módulo que se encarga de devolver por ejemplo una lista de artistas que tengan canciones de un género determinado.

Fichero webfunctions.py

Este fichero contiene los métodos necesarios para generar las respuestas XML a partir de los datos que recibe del módulo mp3functions.py.

Cliente

Fichero jukloud.html

Contiene la plantilla a partir de la cual se construye la interfaz gráfica del reproductor. Incluye código JavaScript para el manejo de eventos.

Camino de datos

1. El servidor construye la interfaz web.
2. El cliente interactúa con la interfaz web, por ejemplo, seleccionando un artista.
3. El cliente detecta el evento del usuario (en este caso la pulsación) mediante JavaScript y lanza varias peticiones AJAX utilizando la función `load()` de jQuery. En este ejemplo se realizan tres peticiones, para actualizar la tabla de álbumes, géneros y canciones respectivamente.
4. El servidor (fichero jukloud.py) llama al método `GET()` de las clases vinculadas a cada una de las URL a las que se ha hecho la petición AJAX. Dichas clases pueden recoger las variables que haya podido pasar el cliente a través de `GET()`.
5. Cada método del servidor llama a la función correspondiente del módulo `webfunctions.py` para obtener la respuesta XML que deberá devolver a la URL que le han solicitado. Por ejemplo, la clase `get_albums_from_artist` recibe por `GET()` el nombre de un artista y llama a la función `getAlbumsFromArtistTable` enviando como parámetro dicho nombre.
6. El módulo `webfunctions.py` recibe las peticiones correspondientes y llama a distintas funciones del módulo `mp3functions.py` para obtener los datos que tendrá que incluir en su respuesta XML. Siguiendo con el ejemplo, la función `getAlbumsFromArtistTable` llama a la función `getAlbumsFromArtist` del módulo `mp3functions.py`.
7. El módulo `mp3functions.py` se conecta a la base de datos y obtiene la respuesta a la petición que se le hace, devolviéndola en forma de lista de tuplas. En este caso, la función `getAlbumsFromArtist` de `mp3functions.py` se conecta a la base de datos y obtiene una lista de tuplas con todos los Álbumes del artista que se le ha pasado como parámetro, devolviendo dicha lista a la función `getAlbumsFromArtistTable` de `webfunctions.py`.
8. El módulo `webfunctions.py` recibe los datos y los transforma de un formato de lista de tuplas a una tabla XML, devolviéndola al método `GET()` de la clase `get_albums_from_artist` del fichero `jukloud.py`.
9. El servidor (fichero jukloud.py) recibe la tabla XML y la renderiza en la URL de respuesta, que en el ejemplo sería `/ajax/getalbumsfromartist`. Esta URL contiene ahora la respuesta AJAX del servidor.
10. La web cliente (fichero jukloud.html) carga la tabla XML de la URL `/ajax/getalbumsfromartist` y actualiza el contenido de la tabla de álbumes con los nuevos datos.

Durante este camino, lo único que la persona que está utilizando el cliente web percibe es que al pulsar en un artista, los campos de álbumes, géneros y canciones se han restringido a aquellos que son del artista que ha seleccionado.

El resto de peticiones siguen básicamente este mismo proceso. En el caso de la reproducción, en la tabla de canciones hay un campo oculto con los paths a cada una de las canciones, por lo que es accesible sin necesidad de hacer una petición AJAX en cada cambio de canción, minimizando el tiempo de respuesta de la aplicación.

Pruebas

Una vez realizada la aplicación se ha comprobado que funcionan sin problemas tanto en Linux como en Windows:

- Es posible ejecutar varias instancias simultáneas del cliente web, ya que el servidor es multihilo, por lo que puede gestionarlas sin problemas.
- Como ya se ha visto, se ha probado el cliente web en varios navegadores, funcionando sin problemas salvo en el caso de Internet Explorer. En el manual de usuario se pueden ver imágenes de la visualización de la aplicación en Internet Explorer 9, Firefox 4 y Safari para iPhone.
- Se ha comprobado que es posible cargar nuevas canciones en la biblioteca sin necesidad de parar el servidor. Las canciones se actualizan en la aplicación web en la siguiente petición AJAX que haga el cliente.
- Se ha comprobado la plantilla de la web para que se adapte correctamente a distintos tamaños de pantalla, como se puede ver en la imagen de la interfaz vista en iPhone (320x480px).

Futuras mejoras

Por falta de tiempo, no se ha podido incluir toda la funcionalidad que el alumno hubiera deseado, en concreto, destaca las siguientes posibles mejoras:

- Compatibilidad con más formatos de archivo, especialmente FLAC y OGG.
- Visualizador con efectos. Existen proyectos de plugins de visualización en HTML5, aunque aún están en fases tempranas de desarrollo.
- Gestión de usuarios y listas de reproducción: Sería deseable establecer un login a los usuarios para restringir el acceso y permitir que los usuarios puedan guardar sus listas de reproducción favoritas.
- Obtención de estadísticas de los usuarios, con el objetivo de crear listas de reproducción dinámicas (temas más escuchados, temas añadidos recientemente, etc.).
- Añadir compatibilidad con Internet Explorer, aunque parece complicado por la dependencia de la web de las nuevas tecnologías.
- Detección de cambios en los archivos de la biblioteca para evitar tener que reinicializarla para conservar la integridad al mover o eliminar archivos.
- Añadir más posibilidades al reproductor, como botones de adelante/atrás o control de volumen.

- Añadir posibilidad de ver carátulas de CD en el reproductor web.
- Mediante las posibilidades de almacenamiento local de HTML5, guardar una copia local de la base de datos para evitar la mayoría de las peticiones AJAX. Añadir un método que compruebe si dicha base de datos está actualizada respecto a la base de datos del servidor.

Bibliografía

- Zeya: Reproductor Web. URL: <http://web.psung.name/zeya/> Última visita:30/05/11.
- Reproductor HTML5 de jQuery. URL: <http://www.jplayer.org/> Última visita:30/05/11.
- Sitio oficial de Web.py. URL: <http://webpy.org/> Última visita:30/05/11.
- Sitio oficial de CherryPy. URL: <http://www.cherrypy.org/> Última visita:30/05/11.
- Sitio oficial de Django. URL: <https://www.djangoproject.com/> Última visita:30/05/11.
- Fundamentos de las páginas web dinámicas. URL: <http://the-geek.org/django-book/chapter3/> Última visita:30/05/11.
- JSON Wikipedia. URL: <http://es.wikipedia.org/wiki/JSON> Última visita:30/05/11.
- SOAP Wikipedia. URL: <http://es.wikipedia.org/wiki/SOAP> Última visita:30/05/11.
- XMLHttpRequest Wikipedia. URL: <http://es.wikipedia.org/wiki/XMLHttpRequest> Última visita:30/05/11.
- AJAX Wikipedia. URL: <http://es.wikipedia.org/wiki/AJAX> Última visita:30/05/11.
- Tag <audio> en HTML5. URL: <http://www.anieto2k.com/2009/07/14/el-elemento-audio-del-html5/> Última visita:30/05/11.
- Tag <audio> en HTML5. URL: <http://sandfly.net.nz/blog/2010/03/the-html5-audio-tag/>
- Implementar un fallback en flash. URL: <http://www.html5rocks.com/tutorials/audio/quick/> Última visita:30/05/11.
- Sistema de plantillas de Web.py. URL: <http://webpy.org/docs/0.3/templetor> Última visita:30/05/11.
- Cómo servir JSON. URL: <http://stackoverflow.com/questions/3513446/webpy-how-to-serve-json> Última visita:30/05/11.
- Ejemplos AJAX. URL: http://foro.elhacker.net/desarrollo_web/libro_de_isitas_sin_que_se_vaya_la_pagina-t314936.0.html;msg1560115#msg1560115 Última visita:30/05/11.
- Tutorial de AJAX con Web.py y jQuery. URL: <http://kooneiform.wordpress.com/2010/02/28/python-and-ajax-for-beginners-with-webpy-and-jquery/> Última visita:30/05/11.
- Tutorial de WebSockets con Python. URL: <http://yz.mit.edu/wp/web-sockets-tutorial-with-simple-python-server/> Última visita:30/05/11.
- Plugin DataTables. URL: <http://www.datatables.net/>
- Tutoriales posicionado CSS. URL: <http://www.barelyfitz.com/screencast/html-training/css/positioning/> Última visita:30/05/11.
- Plantillas CSS. URL: <http://matthewjamestaylor.com/blog/perfect-3-column.htm> Última visita:30/05/11.

Apéndice A: Manual de Usuario

Bienvenido a Jukloud



Gracias por probar Jukloud. Con esta aplicación podrá reproducir su música desde cualquier lugar gracias a su reproductor web. Con él y de una manera muy sencilla podrá escuchar sus canciones desde cualquier navegador, incluyendo los de Smartphones Android e iOS.

Primeros pasos

Para poder utilizar el servidor de Jukloud, el único requisito es tener un intérprete de Python versión 2.6 o superior, e instalar las librerías `web.py` e `eyeD3` que se adjuntan en la carpeta de librerías del trabajo. En sistemas Linux es mucho más sencillo instalarlas desde los repositorios buscándolas por los nombres anteriores.

En sistemas Windows se pueden instalar una vez se tenga el intérprete de Python descomprimiendo las carpetas y utilizando los siguientes comandos desde la consola una vez situados en dentro de cada carpeta:

Web.py

```
$: python.exe setup.py install
```

eyeD3

```
$: python.exe setup.py.in install
```

El servidor de Jukloud

El corazón de Jukloud es su servidor, que generará la interfaz web desde la cual podrá escuchar su música favorita. Dicho servidor funciona perfectamente en Windows y Linux, y debería funcionar sin problemas en otros sistemas operativos como MacOS o cualquiera que incluya un intérprete de Python 2.6 o superior.

Iniciar el servidor de Jukloud es muy sencillo si ha seguido los pasos previos de instalación. En Windows bastará con hacer doble click en el archivo `jukloud.py` o desde la consola ejecutar:

```
$: python.exe jukloud.py
```

En caso de Linux, basta con ejecutarlo desde la terminal con el comando:

```
$: python jukloud.py
```

En sistemas Linux es posible que tenga que dar a Jukloud permisos de ejecución con el comando:

```
$: sudo chmod +x jukloud.py
```

Por defecto, Jukloud utiliza el puerto 8080. Si desea utilizar otro puerto basta con escribirlo como parámetro al iniciar el servidor.

Escaneando la biblioteca de música con Jukloud

Si es la primera vez que utiliza Jukloud o ha añadido recientemente nuevas canciones a su biblioteca, es recomendable actualizar la base de datos de música de Jukloud para que las nuevas canciones estén disponibles desde la web. Para ello basta con poner los ficheros en el directorio static/songs y actualizar la base de datos con el comando:

```
$: python songsdb.py
```

Jukloud buscará nuevos ficheros recursivamente a partir de la carpeta static/songs, por lo que puede incluir todas las carpetas y subcarpetas que desee, así como otros archivos que no sean de audio. Jukloud filtrará automáticamente los ficheros mp3 y detectará ficheros ya incluidos en la base de datos para no añadirlos varias veces a la biblioteca. Es incluso capaz de detectar dos ficheros idénticos en diferentes localizaciones, por lo que minimiza el problema de tener muchas canciones repetidas.

Jukloud se basa en las etiquetas ID3 de los ficheros mp3 para organizarlos y clasificarlos, por lo que es importante que al menos los tags principales sean correctos. En concreto, Jukloud necesitará los tags correspondientes al artista, al álbum, al género, al nombre de la canción y al número de pista dentro del disco. Jukloud acepta las versiones ID3v1, ID3v2, ID3v2.2 e ID3v2.3. Aunque la inmensa mayoría de los mp3 actuales tiene sus etiquetas correctas, es posible que Jukloud no incluya en su base de datos ficheros con etiquetas incompletas para conservar su orden e integridad.

En la versión actual, Jukloud no detecta un archivo mp3 se ha movido o eliminado. Si es así, se debe hacer un escaneo profundo de la base de datos, que puede realizarse con la orden:

```
$: python songsdb.py reset
```

Este escaneo suele durar más que el normal, ya que Jukloud reconstruirá su base de datos completa, por lo que tendrá que añadir todas las pistas de nuevo a la base de datos. Por ello es importante que las pistas que se añadan al directorio static/songs estén ordenadas, lo que evitará tener que moverlas más adelante.

En caso de que una o más pistas se muevan o eliminen del directorio static/songs sin realizar un escaneo profundo de la base de datos hará que dichas pistas no se puedan reproducir desde el cliente aunque se muestren en el mismo.

En caso de que una pista no tenga la información ID3 correcta durante el escaneo hará que Jukloud la descarte y no la añada a la base de datos.

No se deben cambiar el nombre a un fichero mp3 una vez escaneado, ya que luego el servidor no será capaz de encontrarlo para enviárselo al cliente. En caso de que se renombren uno o más ficheros será necesario realizar un escaneo profundo.

Utilizando la interfaz web de Juklound

La interfaz web de Juklound es muy sencilla de usar y a la vez, lo suficientemente potente para realizar casi cualquier actividad habitual de un reproductor de escritorio.

Accediendo a la interfaz desde el navegador

Para acceder a la interfaz web, basta con escribir en la barra de direcciones del navegador la dirección IP donde se encuentre el servidor, seguida de dos puntos (":") y el puerto en el que se está ejecutando el servidor de Juklound, que por defecto es el 8080.

Una vez cargada la interfaz en el navegador tendrá una imagen parecida a la siguiente:



En la parte superior podemos ver el logotipo de Juklound y justo debajo la canción actualmente en reproducción y su artista. Debajo hay tres columnas que permiten seleccionar las canciones de un artista, un álbum o un género determinado simplemente haciendo click con el ratón sobre el campo por el que se quiera filtrar. Por ejemplo, podemos hacer click sobre el artista "Bon Jovi":

jukloud
Bon Jovi - Summertime

Artista

*** Todos los artistas ***

Bon Jovi

Bradley Joseph

Brian Mann

Álbum

*** Todos los álbumes ***

Lost Highway

One Wild Night (live)

Género

*** Todos los géneros ***

Rock

#	Canción	Artista	Álbum	Género
1	Lost Highway	Bon Jovi	Lost Highway	Rock
2	Summertime	Bon Jovi	Lost Highway	Rock
3	(You Want To) Make A Memory	Bon Jovi	Lost Highway	Rock
4	Whole Lot Of Leaving	Bon Jovi	Lost Highway	Rock
5	We Got It Going On	Bon Jovi	Lost Highway	Rock
6	Any Other Day	Bon Jovi	Lost Highway	Rock
7	Seat Next To You	Bon Jovi	Lost Highway	Rock
8	Everybody's Broken	Bon Jovi	Lost Highway	Rock
9	Til We Ain't Strangers Anymore	Bon Jovi	Lost Highway	Rock
10	The Last Night	Bon Jovi	Lost Highway	Rock

⏮ | | 01:08 / 03:20

Tras unos instantes, la interfaz ha cambiado: en el cuadro de álbumes sólo se muestran los álbumes de Bon Jovi, y en el de géneros sólo se muestran los géneros de esos dos álbumes, que en este caso son ambos de Rock. La lista de canciones de la parte superior también ha cambiado y ahora contiene las canciones de esos dos álbumes. Haciendo click en cualquiera de dichas canciones comenzará la reproducción de la misma, continuando con la siguiente tras finalizar, y volviendo al inicio de la lista tras escuchar todas las canciones de la misma.

Aún podemos restringir más la lista de reproducción seleccionando uno de los álbumes de Bon Jovi, por ejemplo One Wild Night:

jukloud

Bon Jovi - (You Want To) Make A Memory

Artista
 *** Todos los artistas ***
 Bon Jovi

Álbum
 *** Todos los álbumes ***
 Lost Highway
 One Wild Night (live)

Género
 *** Todos los géneros ***
 Rock

#	Canción	Artista	Álbum	Género
1	One Wild Night	Bon Jovi	One Wild Night (live)	Rock
2	Bad Medicine	Bon Jovi	One Wild Night (live)	Rock
3	You Give Love A Bad Name	Bon Jovi	One Wild Night (live)	Rock
4	Keep The Faith	Bon Jovi	One Wild Night (live)	Rock
5	Something For The Pain	Bon Jovi	One Wild Night (live)	Rock
6	Just Older	Bon Jovi	One Wild Night (live)	Rock
7	Something To Believe In	Bon Jovi	One Wild Night (live)	Rock
8	Wanted Dead Or Alive	Bon Jovi	One Wild Night (live)	Rock
9	Runaway	Bon Jovi	One Wild Night (live)	Rock
10	In And Out Of Love	Bon Jovi	One Wild Night (live)	Rock

⏮ ⏸ ⏭ ⏹ ⏮ ⏸ ⏭ ⏹

02:37 / 04:38

Como podemos ver, los distintos cuadros se actualizan en tiempo real para incluir por ejemplo en el caso del cuadro de canciones sólo aquellas que pertenezcan al disco *One Wild Night* de *Bon Jovi*.

Si en cualquier momento queremos eliminar los filtros de búsqueda, bastará con hacer click en los campos *Todos los artistas*, *Todos los álbumes* o *Todos los géneros* que se encuentran al inicio de los tres cuadros superiores. Cualquiera de ellos reiniciará los filtros de búsqueda.

En cualquier caso, la lista de reproducción se ordenará alfabéticamente por álbumes y dentro de éstos por número de pista. Se ha elegido esta ordenación por resultar especialmente adecuada en el caso de que la lista de reproducción contenga más de un álbum, y teniendo en cuenta que algunos álbumes, especialmente las recopilaciones pueden no estar en su orden natural si se ordenan por artista.

En la parte inferior se puede ver el reproductor en sí. A la izquierda hay un botón que permite detener y reanudar la reproducción. La barra central indica el progreso de la pista en reproducción y es posible hacer click en la misma para desplazarse por la canción. Por último, en la parte derecha del reproductor se puede ver la duración total de la pista y el tiempo de reproducción actual. Mientras se esté reproduciendo una pista, aparecerá resaltada en la lista de reproducción así como en la parte superior de la interfaz web, debajo del logotipo de Jukloud.

Compatibilidad de Jukloud

Jukloud se ha probado con los navegadores Internet Explorer 9, Chrome/Chromium 11, Firefox 3 y 4, y Safari en su versión para iPhone. Ha funcionado perfectamente en todos excepto en

Internet Explorer 9. Además, en el caso de Safari para iPhone, permite la reproducción en segundo plano, por lo que es posible escuchar música con Jukloud mientras se realizan otras tareas con el iPhone.

Jukloud en Firefox 4

jukloud

Bon Jovi - You Give Love A Bad Name

Artista	Álbum	Género
*** Todos los artistas ***	*** Todos los álbumes ***	*** Todos los géneros ***
Bon Jovi	Lost Highway	Rock
	One Wild Night (live)	

#	Canción	Artista	Álbum	Género
1	One Wild Night	Bon Jovi	One Wild Night (live)	Rock
2	Bad Medicine	Bon Jovi	One Wild Night (live)	Rock
3	You Give Love A Bad Name	Bon Jovi	One Wild Night (live)	Rock
4	Keep The Faith	Bon Jovi	One Wild Night (live)	Rock
5	Something For The Pain	Bon Jovi	One Wild Night (live)	Rock
6	Just Older	Bon Jovi	One Wild Night (live)	Rock
7	Something To Believe In	Bon Jovi	One Wild Night (live)	Rock
8	Wanted Dead Or Alive	Bon Jovi	One Wild Night (live)	Rock
9	Runaway	Bon Jovi	One Wild Night (live)	Rock
10	In And Out Of Love	Bon Jovi	One Wild Night (live)	Rock

03:24 / 03:53

Jukloud en Chrome 11

jukloud

Muse - Falling Away With You

Artista	Álbum	Género
*** Todos los artistas ***	*** Todos los álbumes ***	*** Todos los géneros ***
Bon Jovi	Absolution	Alternativo y Rock Latino
Bradley Joseph	Grand Piano - Narada Anniversary Collection Disc 2	New Age
Brian Mann	Grand Piano - Narada Anniversary	OST
David Arkenstone		Other

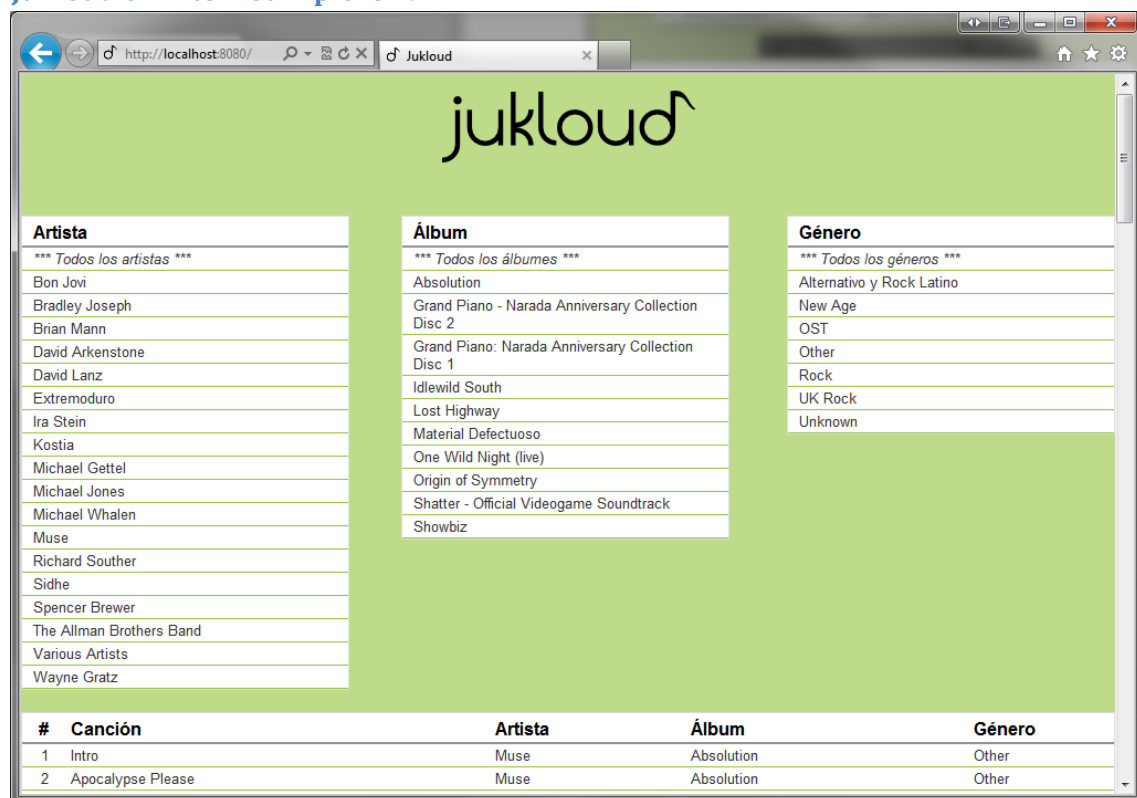
#	Canción	Artista	Álbum	Género
1	Intro	Muse	Absolution	Other
2	Apocalypse Please	Muse	Absolution	Other
3	Time Is Running Out	Muse	Absolution	Other
4	Sing For Absolution	Muse	Absolution	Other
5	Stockholm Syndrome	Muse	Absolution	Other
6	Falling Away With You	Muse	Absolution	Other
7	Interlude	Muse	Absolution	Other
8	Hysteria	Muse	Absolution	Other
9	Blackout	Muse	Absolution	Other
10	Butterflies & Hurricanes	Muse	Absolution	Other

00:01 / 04:42

Jukloud en Safari versión iOS



Jukloud en Internet Explorer 9



Preguntas frecuentes

- ¿Jukloud utiliza flash?(¿Funcionará en mi iPhone/iPad)?

Jukloud no utiliza flash por defecto salvo que el navegador donde se utilice no tenga implementados los elementos de HTML5 necesarios para poder usar dicha tecnología. En este caso, Jukloud tiene disponible una versión del reproductor en Flash para maximizar la compatibilidad con los distintos navegadores web. Jukloud funciona

perfectamente con los dispositivos iOS. En el caso de Android se utiliza la versión flash del reproductor, ya que el navegador por defecto aún no tiene implementada la tecnología necesaria para utilizar HTML5.

- **¿Jukloud es compatible con OGG u otro formato de audio además de mp3?**

Actualmente no. La razones principales son las restricciones temporales del trabajo y el hecho de que el mp3 se considera el estándar *de facto* en cuanto a transmisión de audio vía web. Además, el soporte de los navegadores es bastante irregular en este momento en cuanto a formatos compatibles con HTML5, ya que por ejemplo Safari no soporta el formato OGG, así como Firefox no soporta el formato WAV.

- **¿Cuáles son los requisitos mínimos para ejecutar el servidor de Jukloud?**

En ejecución, Jukloud necesita menos de 15MB en la mayoría de los casos, y la carga de procesamiento es despreciable, aunque aumenta al escanear los ficheros para añadirlos a la base de datos. Cualquier PC actual o de hace unos años tiene potencia de sobra para ejecutar Jukloud.